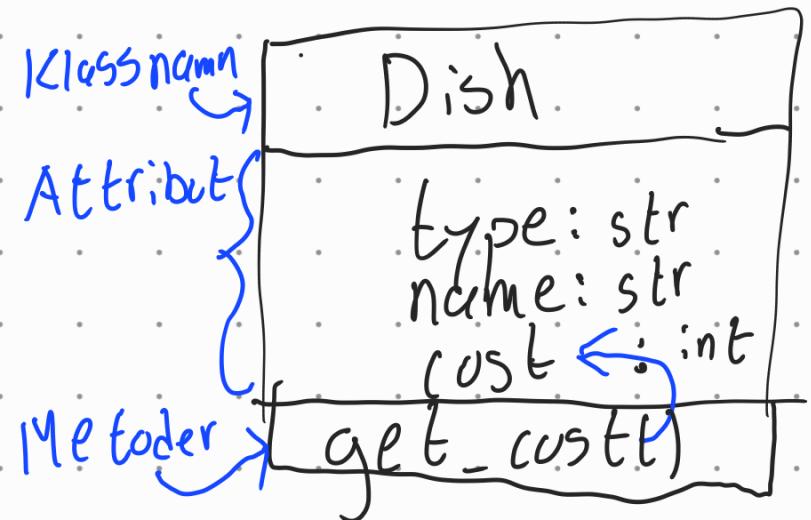


Uppgift 1 (5 min)



(OBS: __init__ ska också skapas)

Ex på hur objekt ska kunna skapas

bread = Dish("starter", "bread", 2)

ratatouille = Dish("main", "ratatouille", 9)

ice_cream = Dish("dessert", "ice cream", 5)

Ex bread.get_cost() => 2

Uppgift 2 (5 min)

Implementera en __str__ metod i Dish. Metoden ska returnera en sträng som beskriver objektet.

Hint: använd objektets attribut strängen

Uppgift 3 (5 min)

Klassnamn

COURSE

Attribut

dishes: Dishes[]

Metoder

-- str --()

Ex.

my-course = Course([bread, ratatouille, ice_cream])

-- str -- ska returna en sträng som beskriver varje maträtt samt ger den totala kostnaden.

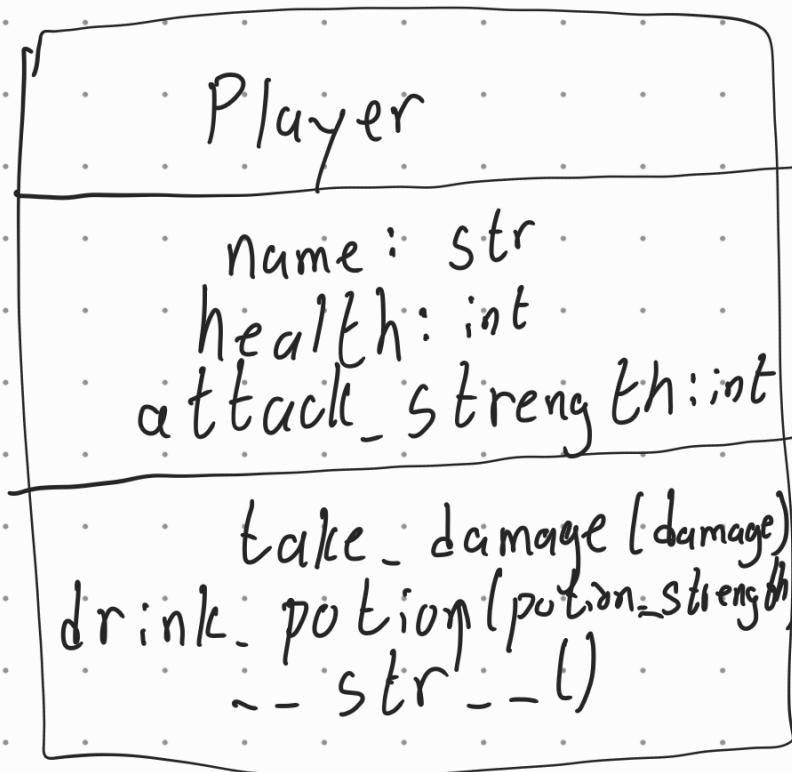
Hint Utnyttja att du redan har skrivit

-- str -- för varje enskild maträtt.

Uppgift 4 (5 min)

-- init -- metoden till Course, kollar så att listan med maträtter har längd 3 samt innehåller en av varje typ maträtt. printar något iömpagt om något inte stämmer.

Uppgift 5 (10 min)



Ex $p1 = \text{Player}(\text{"Gandalf"}, 100, 20)$

`take_damage(damage)` borde subtrahera
damage från spelarens liv och skriva
ut något lämpligt.

`drink_potion(potion_strength)` borde addera
potion_strength till spelarens liv och skriva
ut något lämpligt.

`--str__()` borde returnera en sträng
som beskriver spelaren. De tre attributen
skall finnas med i strängen.

Uppgift 6 (15 min)

Uppg. 10 (15 min)

Duel

current-player: Player
other-player: Player
game-over: Bool

turn()
play()

Ex

duel = Duel(player1, player2)
När duel skapas skall spelarnas __str__-
metoder kallas,

turn() ska föreställa en spelares tur.
current-player ska attackera other-player.
Om other-player dör, sätt game-over
till True.

play() ska kalla på turn() tills spelet
är slut så att vi kan skriva
duel.play()

På vägt nyskapade Duel-objekt för att
starta "matchen".

Väggift 7 (0)

förlagsv.3

Upp till er! Gör spelet mindre förutsägbarat genom att i bland ge potions till spelarna.

impost random

Kan vara bra att använda.

